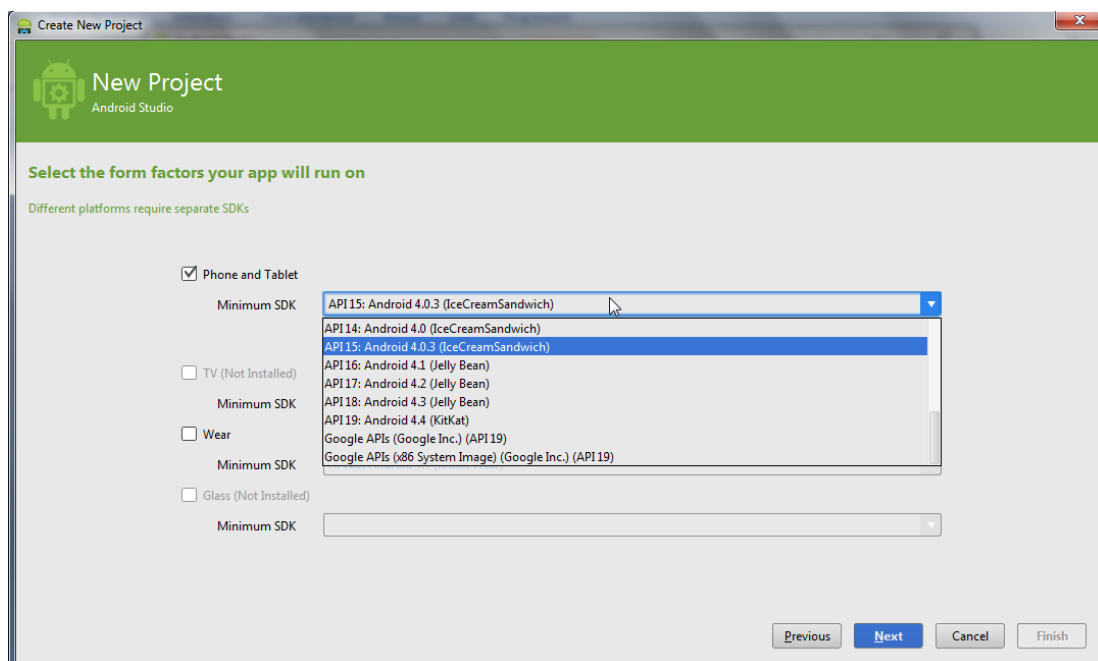
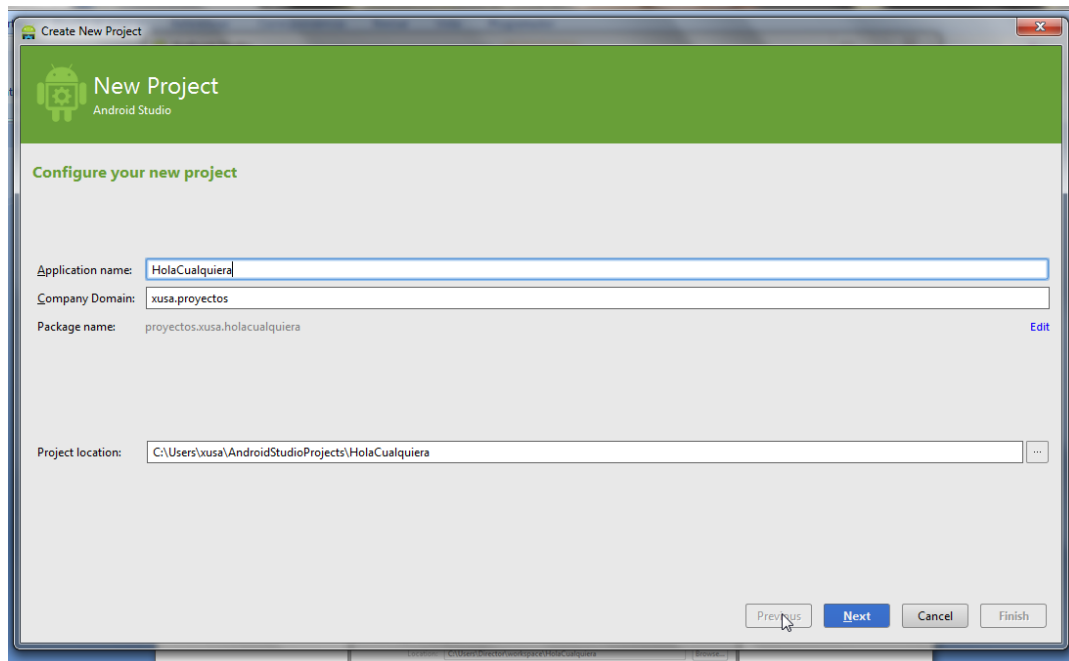
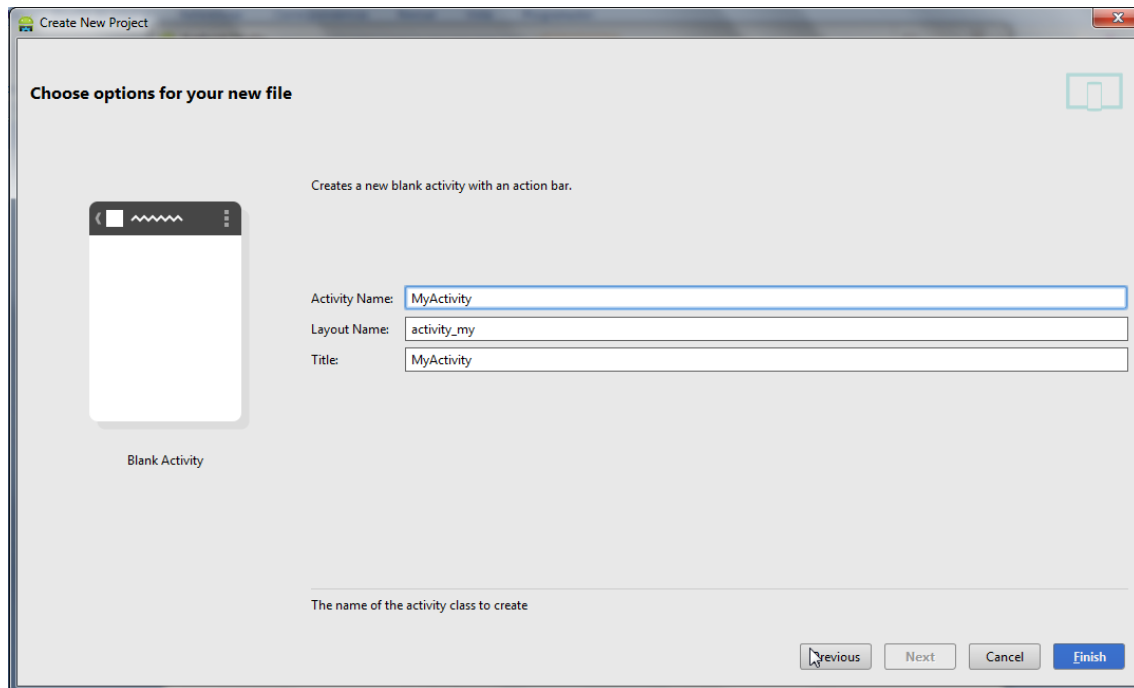


Vamos a crear un nuevo proyecto que se llamará HolaCualquiera. La versión mínima será la 15. La aplicación constará una única pantalla, en la que el usuario introducirá su nombre y cuando el usuario pulse el botón *Saludo*, se visualizará en una campo el saludo, de forma personalizada al usuario, es decir con el mensaje de bienvenida concatenado al nombre del usuario.

PASO1. Creación del proyecto y dar nombre a la Activity





Lo que vamos a hacer a continuación es diseñar nuestra pantalla principal, modificando la que se crea por defecto. ¿Pero dónde y cómo se define cada pantalla de la aplicación? En Android, el diseño y la lógica de una pantalla se encuentran separados en dos ficheros distintos. Por un lado, en el fichero *res/layout/activity_main.xml* tenemos el diseño o apariencia visual de la misma, y por otro lado, en el fichero *src/.../MainActivity.java* encontramos el código java que determina el funcionamiento.

Empecemos por explicar el contenido del fichero *activity_main.xml*. Lo primero que nos encontramos es un elemento *contenedor*, que permitirá la distribución de los controles. Dentro de este hay que incluir tres controles: una etiqueta (*TextView*), un cuadro de texto (*EditText*) y un botón (*Button*). Cada uno de ellos con los siguientes atributos:

android:id, identificador del control, al definirlo como `"@+id/"`, conseguimos que al compilarse el proyecto se genere automáticamente una nueva constante en la clase R para dicho control. Ir al menú *Project* y ejecutar *Clean*. De esta forma se actualiza el contenido de la clase R.

android:text, indicando el texto del control, puede especificarse directamente una cadena o incluyendo alguna de las cadenas definidas para el proyecto en el fichero *res/values/string.xml*.

android:layout_height y *android:layout_width*, que definen las dimensiones del control respect al layout que lo contiene. Podrá tomar los valores *wrap_content* para indicar que las dimensiones del control se ajustarán al contenido del mismo, o *fill_parent* para indicar que el ancho y alto del control se ajustarán al ancho y alto del contenedor respectivamente.

PASO5. Lo primero ha sido cambiar el nombre del string que aparece por defecto, *hello*, del archivo *string.xml*, por *Nombre*. Evidentemente he de modificar también el archivo *activity_main.xml*, indicando el nuevo nombre del string. También el *Button*, para ello primero añadimos un nuevo string llamado *saludar*, con valor *Saludar* y el control *EditText*.

Siendo el código xml:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MyActivity">

    <TextView
        android:text="@string/textoNombre"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

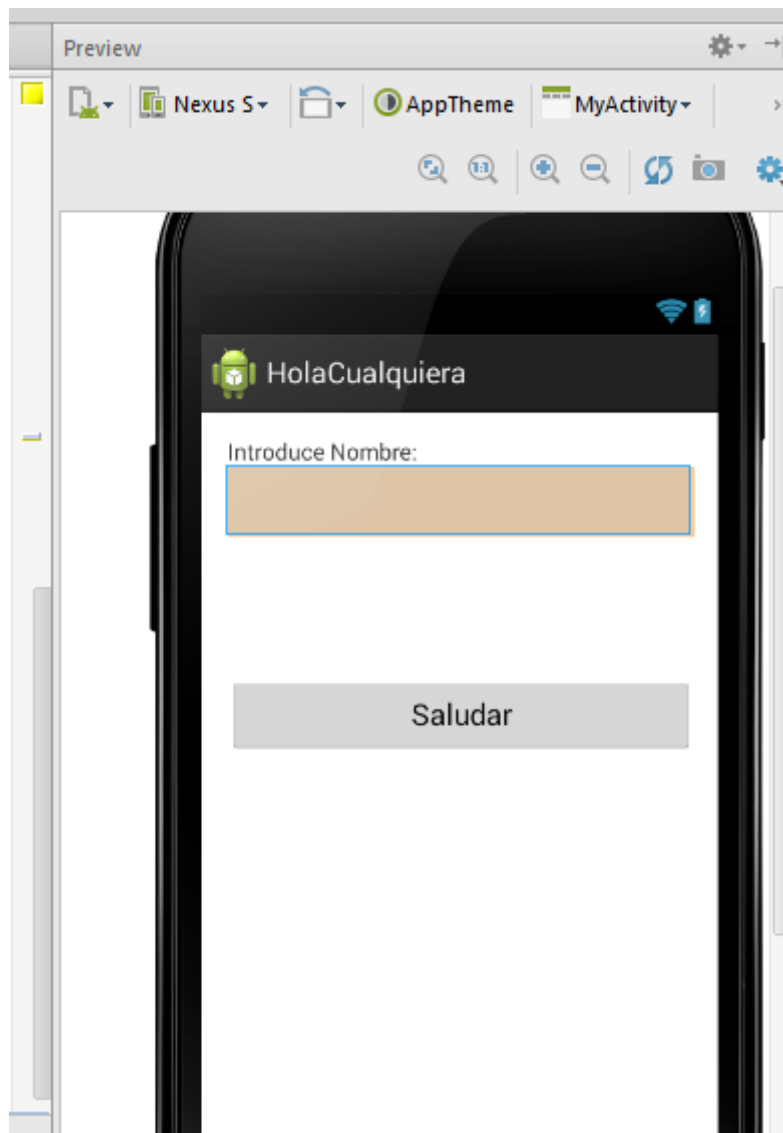
    <EditText
        android:layout_width="match_parent"
        android:layout_height="20pt"
        android:id="@+id/editTextNombre"
        android:editable="true"
        android:layout_gravity="center"
        android:background="#ffffcc9c"
        android:singleLine="false"
        android:layout_below="@+id/textView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/Saludar"
        android:id="@+id/buttonSaludar"
        android:layout_below="@+id/editTextNombre"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="86dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/textView2"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignRight="@+id/textView"
        android:layout_alignEnd="@+id/textView" />

</RelativeLayout>
```

El resultado visual de la primera ventana de nuestra aplicación sería:



PASO2. Vamos a implementar la lógica de funcionamiento de nuestra aplicación. En primer lugar debemos controlar el acceso al cuadro de texto y al botón de nuestra pantalla principal. Para ello necesitamos crear tres variables en la activity y utilizaremos el método *findViewById()* , pasándole el ID de cada control y que están definidos en la clase *R*.

```

public class MyActivity extends Activity {
    EditText ETNombre;
    Button BSaludo;
    TextView ETSaludo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);
        ETNombre=(EditText) findViewById(R.id.editTextNombre);
        BSaludo=(Button) findViewById(R.id.buttonSaludar);
        ETSaludo=(TextView) findViewById(R.id.textView2);

        BSaludo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ETSaludo.setText("hola "+ETNombre.getText()+" , Bienvenido a la aplicación");
            }
        });
    }
}

```

Realizamos los import necesarios (que en Android Studio no serán necesarios)

```

import android.widget.Button;
import android.widget.EditText;

```

...

//Localizar los controles

```

    ETNombre=(EditText) findViewById(R.id.editTextNombre);
    BSaludo=(Button) findViewById(R.id.buttonSaludar);
    ETSaludo=(TextView) findViewById(R.id.textView2);

```

Pasemos a implementar las acciones que han de ejecutarse cuando pulsemos el botón. Para ello implementamos el evento *onClick* del botón. Realmente lo que hacemos es pasar un listener que sobrescribe el método *onClick*.

```

bSaludar.setOnClickListener(new View.OnClickListener() {
    //Es necesario especificar View.OnClickListener() para que no distinga de DialogInterface.OnClickListener()
    public void onClick(View v) {
        ....
    }
}

```

Dentro de este método escribiremos en nuestro *TextView*. Recordar que anteriormente hemos asociado nuestro *TextView* a la variable *tSaludo*. Para recoger el contenido del *EditText*, recurrimos al método *getText()*.

//asociar la cadena al TextView correspondiente

```

ETSaludo.setText("hola "+ETNombre.getText()+" , Bienvenido a la aplicación");

```

Con esto hemos concluido el desarrollo de nuestra aplicación, tan solo nos falta ejecutarla y ver cómo funciona.

AHORA MODIFICA EL CÓDIGO de forma que aparecerá otro botón **Despedida** que nos mostrará un mensaje de despedida similar al saludo, **pero esta vez vamos a derivar nuestra activity de OnClickListener**, en vez de crear un objeto anónimo (es decir utiliza el mismo método *onClick* para los dos eventos).